ENT SENIOR DESIGN PROJECT REPORT

Remote-Controlled Retractable Window

Submitted to

Professor (Dr. David Goodman)
Electrical Engineering Technology Program
Engineering Technology Department
By
Abdulwahab Alabdulwahab
Talal Alzakar
December 4, 2018

ABSTRACT

Every engineering student has a senior project must be completed in order to graduate. Also, all students must complete the projects based on some requirements such as software and hardware parts to accomplish success on their assigned project. Furthermore, the purpose of Remote-Controlled Retractable Window project is to develop a window that is used for homes, companies, and even institutions that has the latest Home Automation technology. Moreover, this paper has a detailed report of the Remote-Controlled Retractable Window project including components, calculations, and design decisions that were made to complete this project in period of two semesters. For instance, calculations for power supplies, microcontroller code, type of motor, type of the remote control, and several objectives to make the window open/close in 10 seconds wirelessly. The results of this project were successful, and the window can be opened/closed in 10 seconds by using RF remote control, stepper motor, two inductive sensors, and PIC microcontroller. Most importantly, the results can help in the development of future Home Automation projects such as a Remote-Controlled Retractable Window.

Error! Unknown document property name.

TABLE OF CONTENTS

ABSTRACT 2
TABLE OF CONTENTS 3
REVISION HISTORY 4
1. INTRODUCTION 5
1. 1 Scope 5
1. 2 System overview 6
Manual and future process 6
Functional Requirements 6
Training Requirements 7
Testing Requirements 7
2. REFERENCED DOCUMENTS 8
3. SYSTEM-WIDE DESIGN DECISIONS 9
3. 1 Hardware 12
3. 2 Software 17
3. 3 Interface 17
4. SYSTEM ARCHITECTURAL DESIGN 21
4.1. System components 21
4.1. 1 Microcontroller Schematic & operation. 21
4.1. 2 Stepper Motor Schematic & Operation. 22
4.1. 3 Stepper Motor Driver Schematic & Operation. 23
4.1. 4 LM2575 Voltage Regulator. 25
4.2. Concept of execution 26
5. User Manual 28
6. Timeline 29
7. CONCLUSIONS AND RECOMMENDATIONS 31
NOTES ..... **Error! Bookmark not defined.**
REFERENCES ..... **Error! Bookmark not defined.**
APPENDIXES 32

Error! Unknown document property name.

REVISION HISTORY

Version	Date	Revised by	Description
1.0	October 9, 2018	Talal Alzakari Abdulwahab Alabdulwahab	1st version
1.1	November 23, 2018	Talal Alzakari Abdulwahab Alabdulwahab	2nd Version
1.2	December 1, 2018	Talal Alzakari Abdulwahab Alabdulwahab	3rd Version: Grammar and spelling Check.

1. INTRODUCTION

The project is about Retractable Windows that is controlled by a remote controller which can be used for homes, businesses, or apartments. This is an industrial project which has different requirements than the senior class's requirements. Both the quality and the cost are the most concerns in this project. The project must be robust enough during the test before it delivers to the sponsor who is Terry Walden. Furthermore, this project is a group of Mechanical and Electrical Engineering students. In this paper, the electrical aspect of the project will be discussed. Moreover, this project has been developed based on the requirements of the sponsor. The requirements are the window must be opened and closed in 10 seconds, it should be powered from AC outlet, and controlled by a remote control. The design of this project consists of electrical motor, power supply, sensors, microcontroller and remote control.

1. 1 Scope

The scope of this project gives a full identification of the scope of this project is to design and create a Remote-Controlled Retractable Window.

- The motor will allow the mechanical components inside the pocket window to move which will open or close window without manual force.
- The motor speed will take seconds for opening and closing.
- The system will insure that appropriate safety features are employed.
- A switch will allow the user to shut down the power manually.
- The window will be able to be opened or closed manually.
- Remote push-buttons must send two functions to microcontroller.
- The motor will stop when the window is fully closed or fully opened.

Error! Unknown document property name.

1. 2 System overview

Manual and future process

Table 1 - Manual and future process

Manual Process	*Proposed Remote-Control Process*
Decide to open or close the window.	Decide to open or close the window.
Stands and see where the window is located.	Discovered where the remote control is.
Walk into the window.	Get the remote control.
Hold the window's holder.	Press the Remote-control's buttons to allow the motor to move either direction which will cause the window to open or close depending on the button pressed.
Pull the window to left or right for opening or closing.	Watch the window while it is opening or closing

Functional Requirements

Table 2 - Functional Requirements

Error! Unknown document property name.

#	Source	Requirement	Priority
1	Motor Speed	Given the provided motor with a rotational speed of 500-900 RPM the window will be able to fully open or close within 6-10 seconds.	**High**
2	Motor movements	The window must move in both directions to be able to open and close.	**High**
3	Switch	A switch should be added to the system to turn the power on and off for safety conditions.	**High**
4	Fuse	A fuse should be added to the application to shut down the power when the application produces more current than it should. The home owner or window technician must be able to replace the fuse without damage to the wall or window.	**High** **High**
5	Microcontroller	Turn the motor on when receive signal.	
6	Remote control	The remote control will be able to open and close the window.	**High**
7	Sensor	A Sensor will be added in the window to stop the window when it is fully closed or fully opened.	

Training Requirements

Table 3 - Training Requirements

#	Source	Requirement	Priority
1	Installation	Limited training provided to the sponsor at demonstration.	**High**

Testing Requirements

Table 1 - Testing Requirements

Error! Unknown document property name.

#	Source	Requirement	Priority
1	Motor Speed	Turn the Power on, make sure the window is fully closed, press the opening push-button, and use a timer to count the time it takes for fully opening.	**High**
2	Motor movement	Press the push-buttons to open or close the window.	**High**
3	Sensor	The window must stop when the window is fully closed or fully opened.	**High**

2. REFERENCED DOCUMENTS

This section shall list the number, title, revision, and date of all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.

Table 4: Reference Documents

Title	Source	Comment
Voltage Regulator	LM2575 Datasheet	Operation and condition
Stepper Motor Driver SR-4	SR-4 Datasheet	Operation and condition
Pic Microcontroller	PIC16F87XA	Operation and condition

3. SYSTEM-WIDE DESIGN DECISIONS

First, there are five pictures at the bottom that shows how the system must have. Also, what are the system components such as the motor types, stepper motor driver types, stepper motor types, microcontroller types, and most importantly remote types. Each one of them will show how we went through several steps and conditions to insure the right and accurate selection. Most importantly, what will be more satisfied for the sponsor of the project.

Solution Selection Matrix

	Very Low (less good)		Moderate		Very High (best)		
	1	2	3	4	5		
Potential Solution (Provide Brief Description)	**Efficiency**	**Positive Customer Impact(1= less controllable & 5= more controllable)**	**Cost to Implement (1 = $$$ & 5 = $)**	**Torque and speed(1=don't Match& 5 = match**	**Time to Implement (1 = Long 5 = Quick)**	**Total Score**	**Implement? Yes/No**
Weighted Criteria	5	10	8	10	5		
DC Motor	4	3	5	4	4	*150*	No
AC motor	4	3	4	3	4	132	No
Stepper motor	5	5	3	5	2	159	Yes

The final score show that the stepper motor is the best selected type of motor for this project. The most important factors for this project are Positive Customer impact. A stepper motor is more precise than the other types of motor because it moves 1.8 degree each steps.

Solution Selection Matrix for steper motor							
	Very Low (less good)		Moderate		Very High (best)		
	1	2	3	4	5		
Potential Solution (Provide Brief Description)	**Efficiency**	**Size (1= beigger then 4 'x 4" & 5 = less then)**	**Cost to Implement (1 = $$$ & 5 = $)**	**Torque and speed(1=less then needed& 5 = match the needed)**	**Time to delivered (1 = Long 5 = Quick)**	**Total Score**	**Implement? Yes/No**
Weighted Criteria	5	7	8	10	10		
34HS59-5004S	5	5	5	4	3	*170*	No
ML23HSCP4300	5	5	4	5	4	182	Yes
Autonics A4K-M245 Stepper Mtr	5	5	3	5	3	164	No

These types of stepper motor are sharing the same feature, but the cost and the time are different. ML23HSCP4300 was selected because it cost less than option one, and it takes less time to be delivered than option 3. In future improvements of this project option 3 might be selected when the time is not matter.

Error! Unknown document property name.

Solution Selection Stepper Motor Drive							
	Very Low (less good)		Moderate		Very High (best)		
	1	2	3	4	5		
Potential Solution (Provide Brief Description)	Efficiency	Size (1= beigger then 4 'x 4" & 5 = less then)	Cost to Implement (1 = $$$ & 5 = $)	Torque and speed(1=less then needed& 5 = match the needed)	Time to delivered (1 = Long 5 = Quick)	Total Score	Implement? Yes/No
Weighted Criteria	5	10	8	9	10		
SR8-Plus	4	3	4	5	1	*137*	No
SR4-Plus	4	3	5	1	5	149	No

SR8-Plus is a better option than SR4-Plus, But SR4-Plus was selected because of the time SR8-8 would take to be delivered. SR8-Plus was not in stock, so SR4 was selected in this project as alternative option.

Potential Solution (Provide Brief Description)	Efficiency	Customer use (Hard = 1 & Easy = 5)	Cost to Implement (1 = $$$ & 5 = $)	Range (1 = Bad & 5 = Good)	Time to implement (1 = Long 5 = Quick)	Total Score	Implement? Yes/No
Weighted Criteria	5	6	8	10	5		
IR Remote Control	2	4	5	1	2	*94*	No
RF Remote Control	5	5	4	5	4	157	Yes

RF remote controller is selected because of its range of operation. Also, IR remote must be aimed to the receiver to receive the signal which consider as low efficiency.

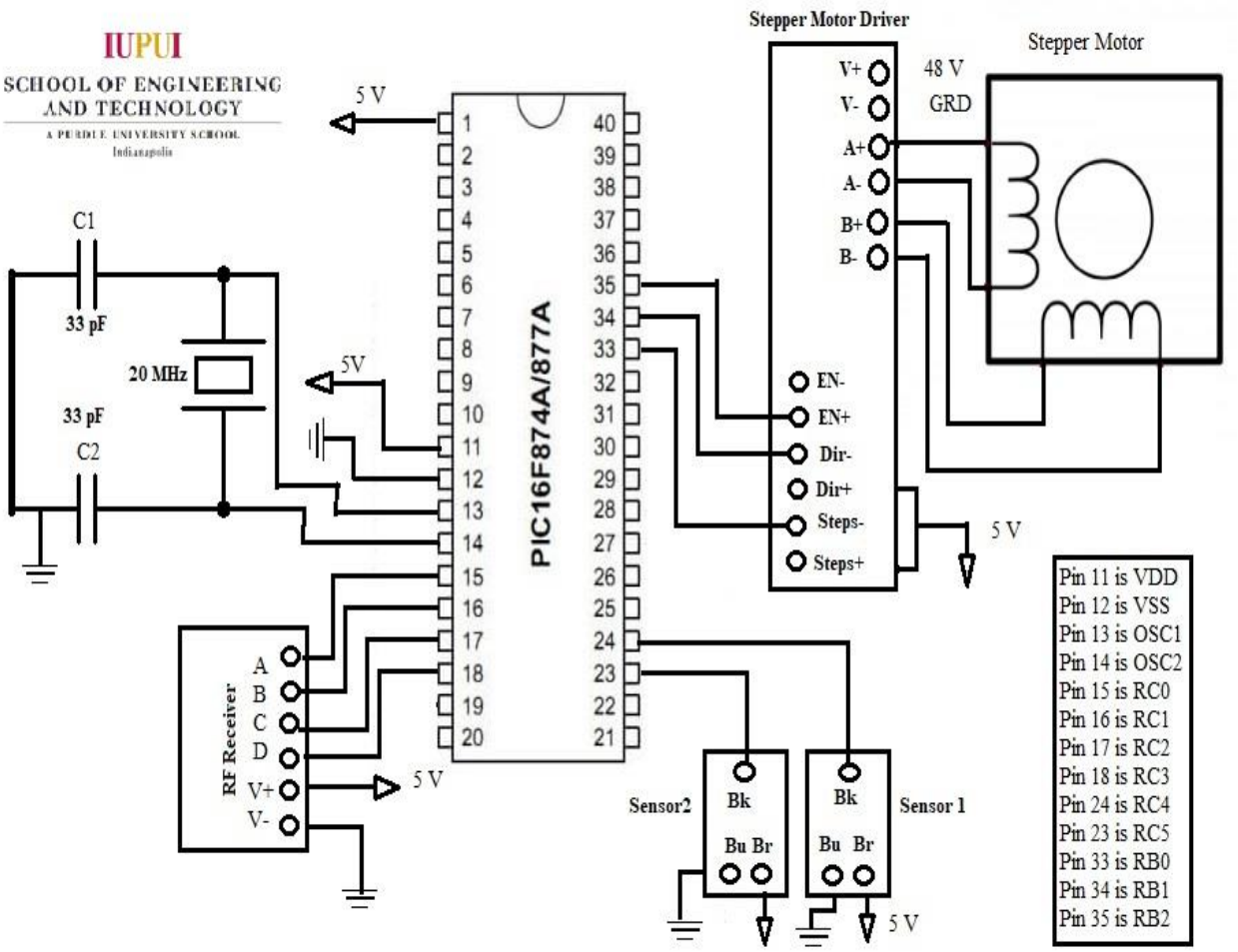
Potential Solution (Provide Brief Description)	Efficiency	Robust (1= Not Robust & 5 = High Robust)	Cost to Implement (1 = $$$ & 5 = $)	Time to implement (1 = Long 5 = Quick)	Total Score	Implement? Yes/No
Weighted Criteria	5	6	8	5		
Arduino Microcontroller	**5**	**1**	**2**	**2**	*57*	**No**
PIC Microcontroller	**5**	**5**	**4**	**4**	107	**Yes**

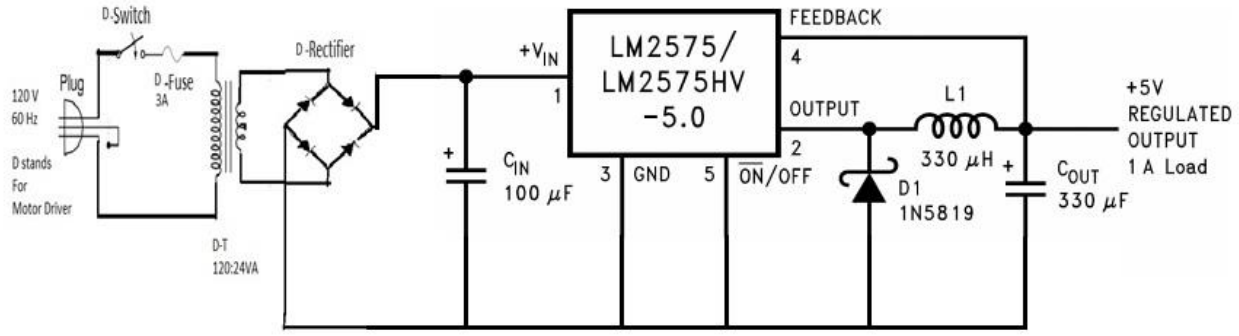
Pic Microcontroller is used over Arduino because it is more robust and better use for industry.

3. 1 Hardware

This Figure shows all the hardware components used to implement this project which are PIC microcontroller, two inductive sensors, stepper motor, stepper motor driver, RF receiver, and the DC voltage.

Error! Unknown document property name.





Error! Unknown document property name.

Error! Unknown document property name.

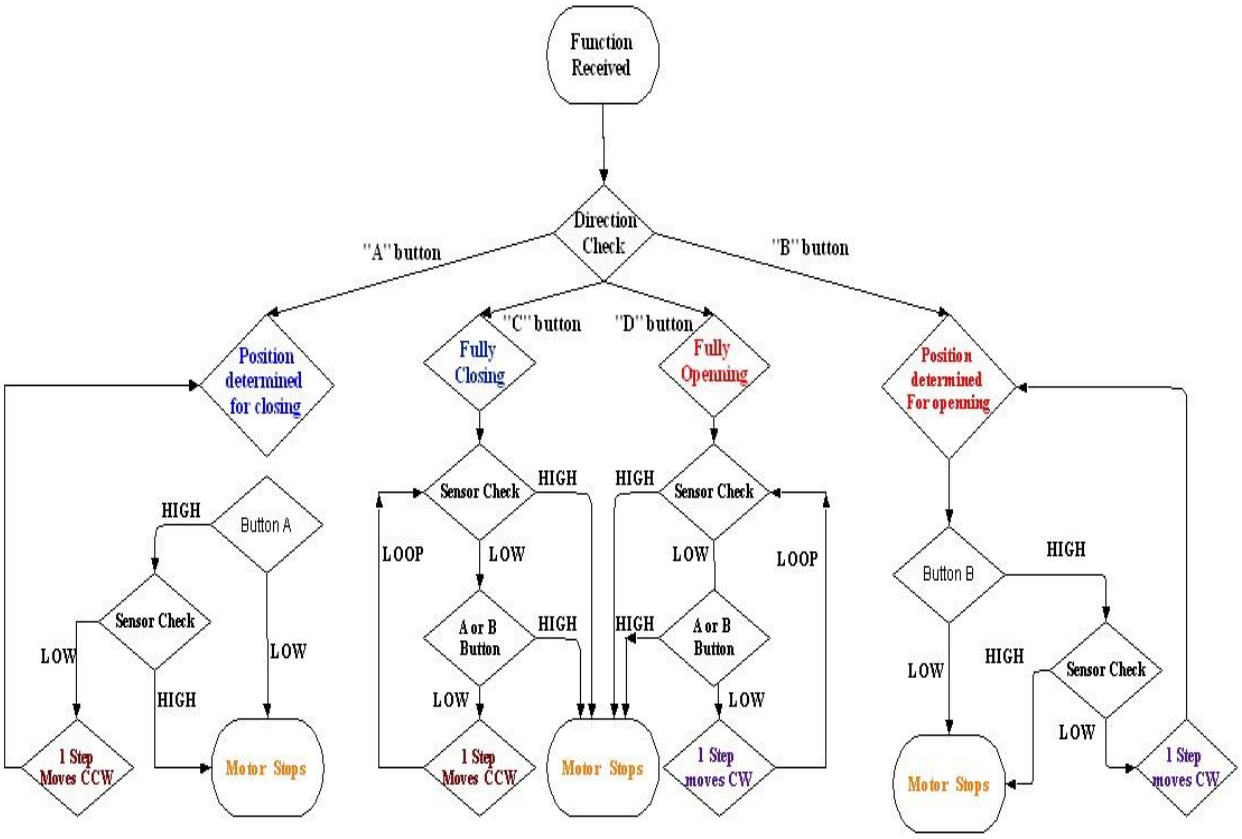
Error! Unknown document property name.

Bill of Materials - Remote-controlled Retractable Window

ITEM NO.	Value.	REC. No.	MFR Name / PART NO.	Cost	QTY.	DESCRIPTION
1	3A	Fuse	6F059	$ 8.00	1	3A Fast Acting Glass Fuse with 250VAC Voltage Rating; AGX Series
2	1A	Fuse	AGC-1-R	$ 4.80	1	1A Fast Acting Glass Fuse with 250VAC Voltage Rating; AGC Series
3	NA	Pic-Microcontroller	PIC16F877A-E/P	$ 6.53	1	8-bit Microcontrollers - MCU 14KB 368 RAM 33 I/O
4	20 MHz	Frequency Oscillator	2039001821	$ 1.00	1	Oscillator; Crystal; 20MHz; HC-49SD;
5	33p	33pF Ceramic Capacitor	561R10TCCQ33	$ 0.45	2	Capacitor; Ceramic; Cap 33 pF; Tol 5%; Vol-Rtg 1000 VDC; Radial
6	NA	Stepper Motor Driver	[SR8-Plus]	$ 119.00	1	DC Input, DIP and Rotary Switch setting, Pulse & Direction Control Drive
7	NA	Stepper Motor	[ML23HSCP4300]	$ 98.70	1	Stepper motor, High torque, Low inertia, Fast raising speed, 2 phase, 1.8°, NEMA23
8	NA	Remote Controller and Receiver	700-10016	$ 20.99	1	RF Modules Key Fob Remote
9	,48 V	Power Supply	B07FQ72CKW	$ 16.99	1	AC-DC 48V 4A Isolated Power Supply Module AC 110V 100-240V to 48V 4A
10	5 V	Voltage Regulator	LM2575T-5.0/NOPB-ND	$ 2.68	1	IC REG BUCK 5V 1A TO220-5
11	100 uF	Capacitor	493-1148-ND	$ 0.52	1	CAP ALUM 100UF 20% 100V RADIAL
12	N/A	Diode	1N5819A0GCT-ND	$ 0.30	1	DIODE SCHOTTKY 40V 1A DO204AL
13	330 uH	Inductor	AIUR-06-331K-ND	$ 1.20	1	FIXED IND 330UH 1.3A 394 MOHM TH
14	330 uF	Capacitor	1189-1891-ND	$ 0.50	1	CAP ALUM 330UF 20% 35V RADIAL
15	5V	Power Supply	Walfrontbozh23ydrk	$ 17.20	1	AC-DC Isolated Switching Power Supply Module Input 85V-264V Output 5V 2A 10W
16	NA	Proximity Sensor	41111926	$ 15.99	2	Normally open) LJ12A3-4-Z/BX-5V cylinder inductive proximity sensor switch work voltage 5VDC
Total cost		$ 315.30				

3. 2 Software

This figure shows a software flow chart of this project which consist of four different option that the user can select.

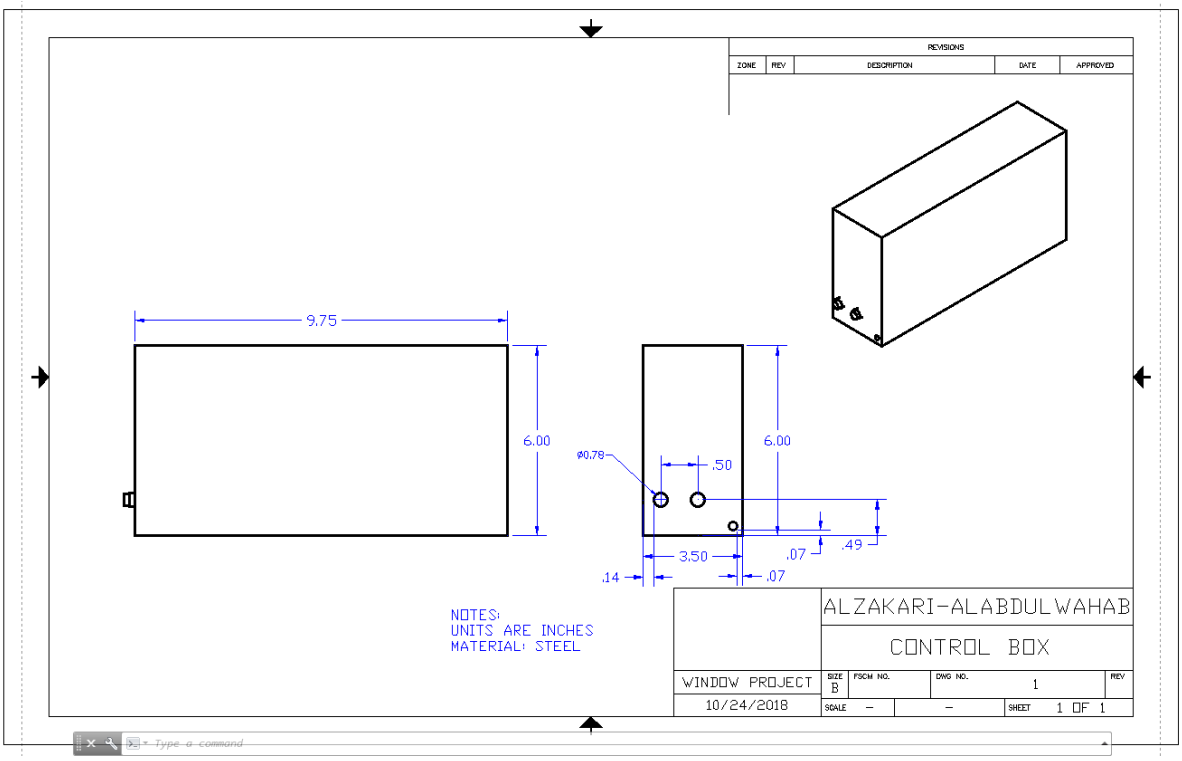


3. 3 Interface

At the bottom, there're two pictures one for the remote control and one for the control box. The remote control shows how to operate it by using four buttons, hold A to open, hold B to close, one press C for fully open, one press D for fully closed. Furthermore, by pressing A or B while window is moving, the window will stop immediately. Also, the control box has all the electrical components such as microcontroller, RF receiver, power supplies, and motor driver.

Error! Unknown document property name.

Error! Unknown document property name.



4. SYSTEM ARCHITECTURAL DESIGN

4.1. System components

4.1. 1 Microcontroller Schematic & operation.

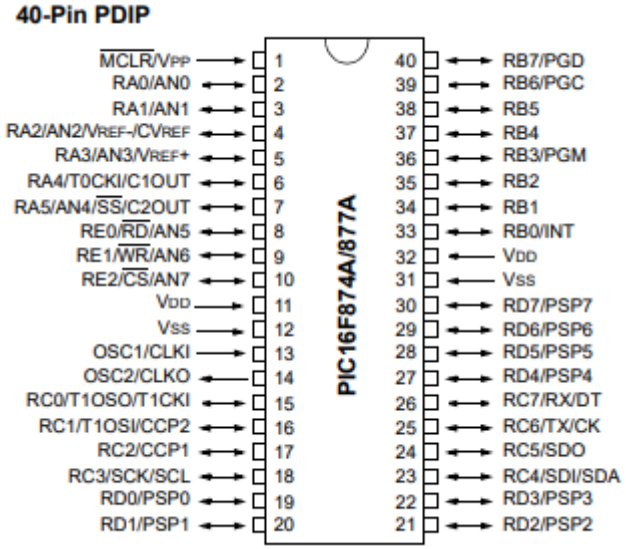


Figure 6

Pic Microcontroller is the smallest microcontrollers that can be programmed to carry out a huge range of tasks. The main features of PIC microcontrollers are RAM, flash memory, Timers/Counters, EEPROM, I/O Ports, USART, CCP (Capture/Compare/PWM module), SSP, Comparator, ADC (analog to digital converter), PSP(parallel slave port), LCD and ICSP (in circuit serial programming). For this project, the first 3 bits of PortB is set as output. Pin 33 is connected to Steps- of stepper motor driver, Pin 34 is connected to Dir- of stepper motor, and Pin 35 is connected to EN+ of stepper motor driver. Also, PortC is set as input. Pins 15 to 18 is

connected to the four channel of the remote controller and pins 24 and 25 are connected to Sensor1 and Sensor2.

4.1. 2 Stepper Motor Schematic & Operation.

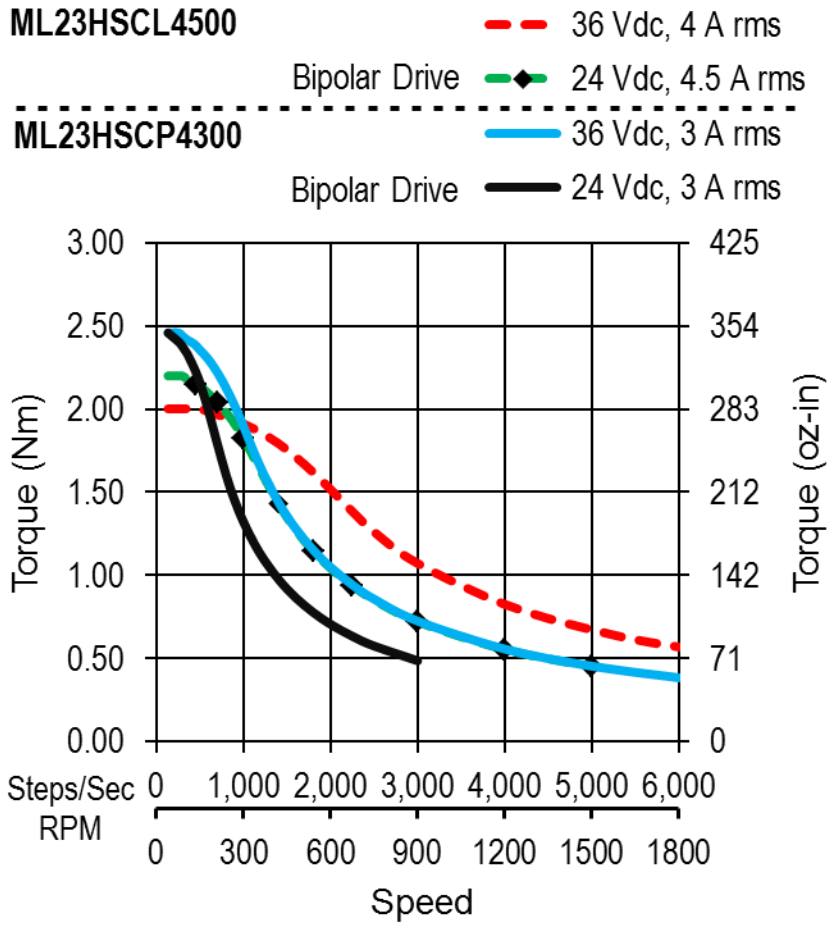


Figure 6

Torque

Ft= Fa+Ff+Fe

Fa= w/g (a/12) = 12lb/32.2 X (0.155in/s^2/12)= 0.0048

Fe = 12 lbs.

Ff=F x u=12 x 0.25 = 3 lbs.

Ft =15.0048

T (torque) = Ft (Lead/2bi (efficiency) = 15.0048(0.2/2pi x0.15) = 3.18 in.lbs

Speed is between 500 Rpm to 900 Rpm to be in the range of 5 to 10 seconds for opening and closing.

4.1. 3 Stepper Motor Driver Schematic & Operation.

a) Stepper Motor Driver Schematic & operation.

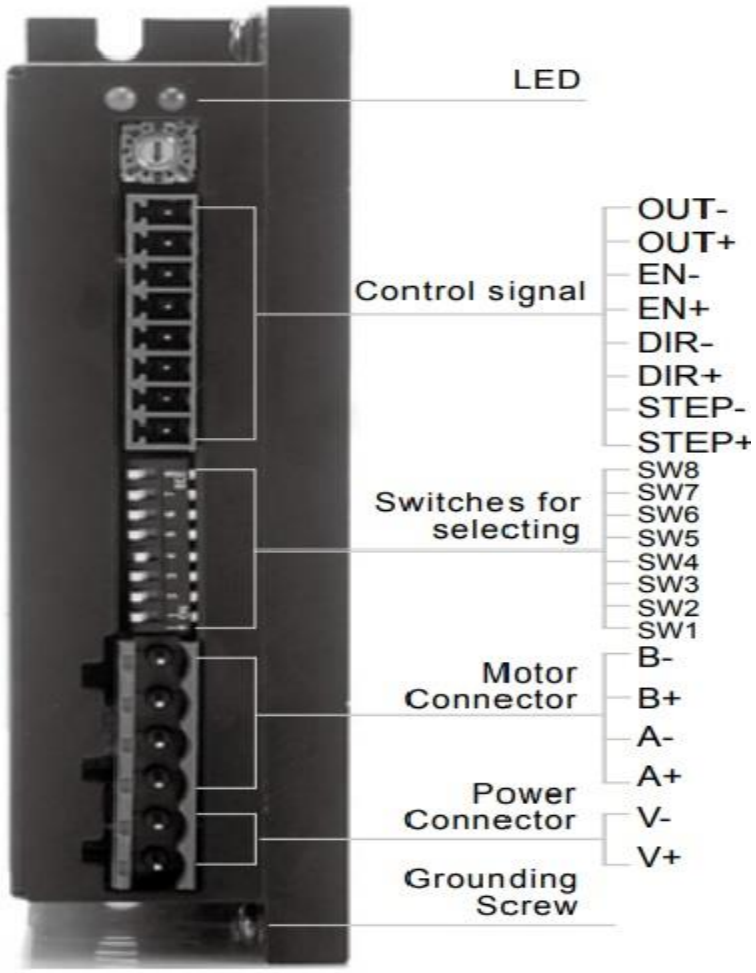


Figure 7

The SR4-Plus drivers include two high-speed inputs called STEP and DIR. They accept 5 to 24 volt single-ended or differential signals, up to 2 MHz typically these inputs connect to a Pic-Microcontroller which provides steps and direction command signals. Also, A+, A-, B+, B- are connected to the motor wires to rotate the motor. More, V- and V+ are the input power which accepts a voltage between 25.0 V and 48.0 V. PWM is used to control the Step- to rotate the motor and Step + is connected to 5.0 V. Dir- is used to control the direction of the motor, it moves CW when Dir- is low and CCW when Dir- is high and

Dir= is connected to 5.0 V. En pins are used to enable all the MOSFET inside the stepper motor driver which will make it in sleep mode.

4.1. 4 LM2575 Voltage Regulator.

The LM2575 series of regulators are monolithic • 3.3V, 5V, 12V, 15V, and Adjustable Output integrated circuits that provide all the active functions Versions for a step-down (buck) switching regulator, capable of • Adjustable Version Output Voltage Range, driving a 1A load with excellent line and load – 1.23V to 37V (57V for HV Version) ±4% regulation. These devices are available in fixed output Max Over voltages of 3.3V, 5V, 12V, 15V, and an adjustable output version.

- **D-rectifier**

A bridge rectifier is four diodes used in a bridge circuit configuration which provides the same output polarity for either input polarity. It is used to convert an alternating current (AC) input into a direct current (DC) output.

- **D-Fuse**

A fuse is used to protect the device from having more current than the device can handle. Also, D-Fuse is 1.00 amps because the maximum current system is 1.00 amps.

- **D-T**

D-T is a transformer that is used to reduce the voltage. According to (LT2575) datasheet, the input voltage is between 7 volts and 40 volts to output 5.0 volts

24.0 rms * 2^ ½ = 33.8Vp

33.8 Vp - 1.40 Vp (voltage drops across diodes) = 32.44 Vp

32.44 Vp / 2^ ½ = 23.0 Vrms

(23.0 Vrms*2 *2^ ½)/ Pi = 20.6 Vdc

Therefore, the best value for the transformer is 120V:24V.

4.2. Concept of execution

First, the program should identify the signal received from remote control receiver and determine channel that is selected. There are four channel which are “A”, “B”, “C”, and “D”. “A” and “B” are Position determined for closing and opening which means when “A’ or “B” is high the motor will move one step and when they are low, the motor will stop moving. “C” and “D” are fully opening and fully closing which means the window will be fully opening or fully closing once “C” or “D” gets high. Also, when “A” or “B is high it will toggle “C” or “D” and it will stop moving the motor.

```
TRISB = 0x00; // Set PortB as output to connect them to Motor
        // RB0 to control the steps.
        // RB1 to control the direction, High moves CW and Low moves CCW
        // RB2 to control the Enable, High to stop the motor, Low will move the motor.
 PORTB = 0x00; // Initialized to be LOW
 TRISC = 0x01; // Set PortC as Input, Portc0 is A channel of re`1mote.
        // Set PortC as Input, Portc1 is B channel of remote.
        // Set PortC as Input, Portc2 is D channel of remote.
```

```
        // Set PortC as Input, Portc3 is C channel of remote.
        // Set PortC as Input, Portc4 is to check the sensor when the window is fully opened.
        // Set PortC as Input, Portc5 is to check the sensor when the window is fully Closed.
 PORTC = 0xF0; // set the first four bits of PortC as Low.
        // set the last four bits of PortC as High. Because the sensor is always high and it gets high once
it detects.
```

The following code used if else statement to first check if channel A is high or not, and if the sensor is high or not. Also, the while loop is used for infinity looping unless one of the condition falls. Channel B uses the same technique but different Port.

```
 if (RC0 == 1) // If we recieve a function from A channel
 {
  while(RC0 == 1) // While A channel is pressed.
  {
     if(RC4 == 1)   // check sensor.
    {
```

The following code used If statements to check if the sensor is high or not and another if statement to check if Channel “A” or “B” is low or not. A while loop will loop 20000 times once channel C is high. The loop will continue looping until one of the statements falls which are channel “A” or “B” gets high and Sensor gets Low. Channel D used the same technique but different Port.

```
if (RC2 == 1) // if channel C is pressed.
  {
    int Count1 = 0;
    while (Count1 <20000)// while loop for fully opening
    {
        if(RC4 ==1&& RC1 ==0 && RC0==0) // check sensor and “A” “B” channel.
        {
```

Error! Unknown document property name.

5. User Manual

IUPUI

SCHOOL OF ENGINEERING AND TECHNOLOGY

A PURDUE UNIVERSITY SCHOOL

Indianapolis

Remote Features:

- Signal indicator LED.
- Up to 50 foot line-of sight range.
- The user can Open or Close the window while he/she is in other room.

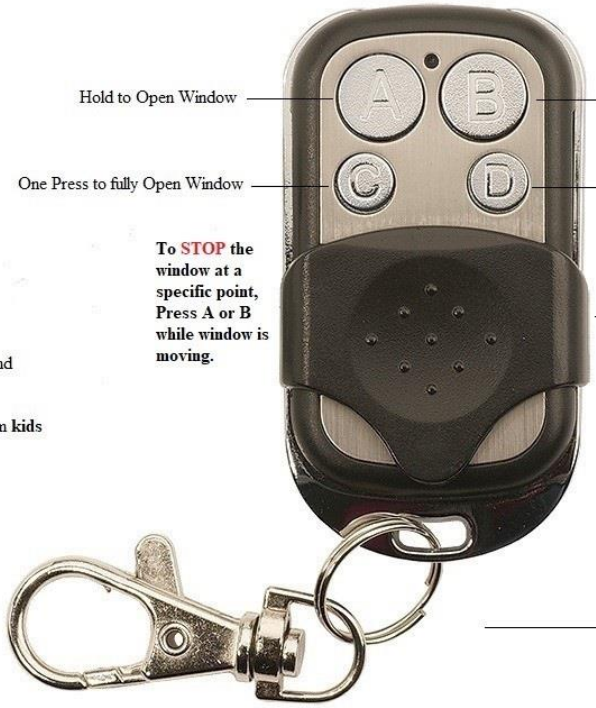
Safety Instructions:

- Make sure to Close the **Remote Protection** at end of day to avoid any damages.
- Make sure to leave the Remote Control away from **kids zone** to avoid any harm.
- Make sure to leave the Remote Control away from your **Pets.**

Remote Battery:

- The remote comes with the battery pre-installed, and it may be replaced if needed. To access the battery compartment, remove the three tiny Phillips head screws on the back of the remote, and remove the back cover.

Operating Time:
10 seconds.

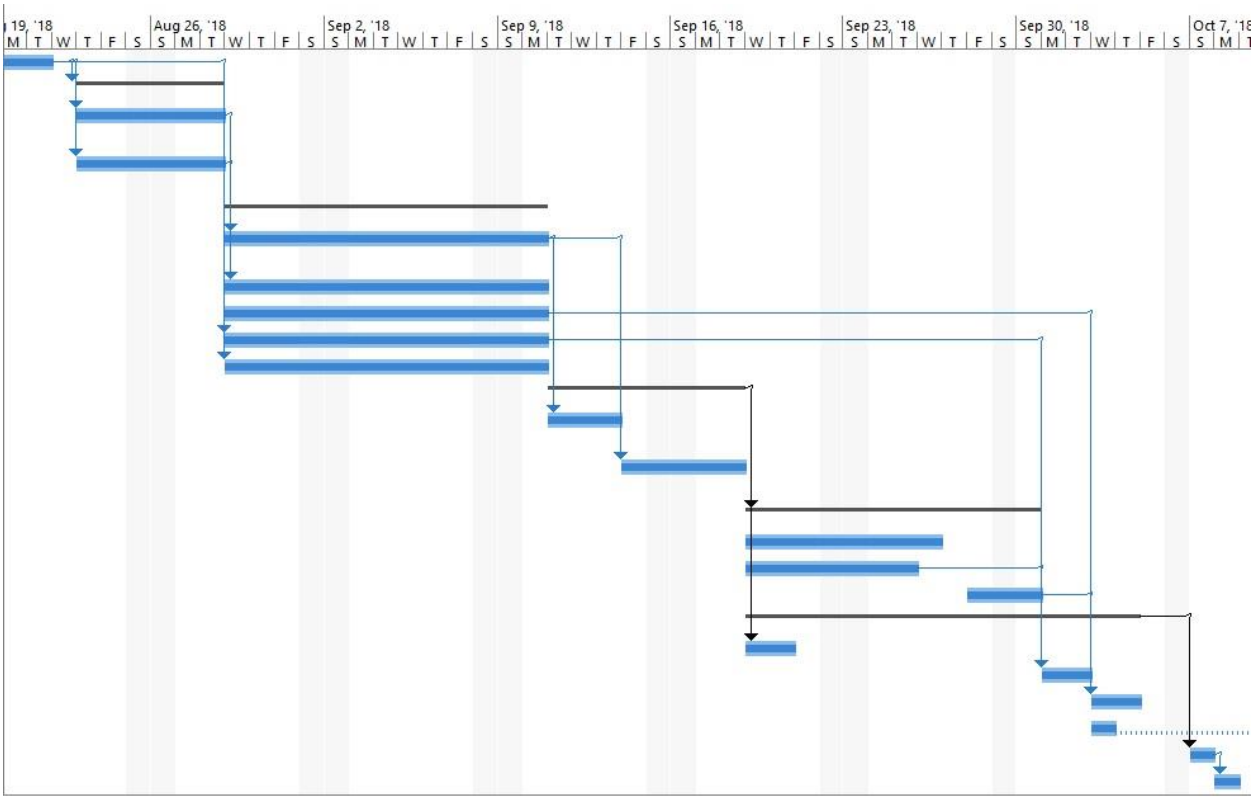


Error! Unknown document property name.

6. Timeline

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		Final Design Changes	2 days	Mon 8/20/18	Tue 8/21/18	
2		**Design Board**	**4 days**	**Thu 8/23/18**	**Tue 8/28/18**	**1**
3		Design 25.0 V P.S board	4 days	Thu 8/23/18	Tue 8/28/18	1
4		Design 5.00 V P.S board	4 days	Thu 8/23/18	Tue 8/28/18	1
5		**Order Componenta**	**9 days**	**Wed 8/29/18**	**Mon 9/10/18**	
6		Order Power Supply's component	9 days	Wed 8/29/18	Mon 9/10/18	3,4
7		Order P.S Boards	9 days	Wed 8/29/18	Mon 9/10/18	3,4
8		order Sensor	9 days	Wed 8/29/18	Mon 9/10/18	
9		Order Remote-contro	9 days	Wed 8/29/18	Mon 9/10/18	1
10		Order Arduino	9 days	Wed 8/29/18	Mon 9/10/18	1
11		**Soldering**	**6 days**	**Tue 9/11/18**	**Tue 9/18/18**	
12		Soldering 25 V Power Supply	3 days	Tue 9/11/18	Thu 9/13/18	6
13		Soldering 5.0 V Power Supply	3 days	Fri 9/14/18	Tue 9/18/18	6
14		**Software**	**9 days**	**Wed 9/19/18**	**Sun 9/30/18**	**11**
15		Stepper motor coding	6 days	Wed 9/19/18	Wed 9/26/18	
16		Remote control codin	5 days	Wed 9/19/18	Tue 9/25/18	
17		Sensor Coding	2 days	Fri 9/28/18	Sun 9/30/18	
18		**Test Components**	**13 days**	**Wed 9/19/18**	**Thu 10/4/18**	
19		Test Power Supplies	2 days	Wed 9/19/18	Thu 9/20/18	11
20		Test Remote control	2 days	Mon 10/1/18	Tue 10/2/18	9,16
21		Test Sensor coding	2 days	Wed 10/3/18	Thu 10/4/18	17,8
22		Test Stepper Motor	3 days	Wed 10/3/18	Thu 11/15/18	
23		Gathering components	1 day	Sun 10/7/18	Sun 10/7/18	18
24		Demo Day	1 day	Mon 10/8/18	Mon 10/8/18	23

Error! Unknown document property name.



7. CONCLUSIONS AND RECOMMENDATIONS

In conclusion, Remote-Controlled Retractable Window is a project that was sponsored by a Terry Walden who owns a company for home remodeling. The purpose of Remote-Controlled Retractable Window project is to develop a window that is controlled wirelessly with a remote control. It is used for homes, companies, and even institutions that has the latest Home Automation technology. The time operation required by the sponsor is 10 seconds for opening and closing. The project went through several tests on different components before it was done. For instance, troubleshooting the PIC microcontroller circuit through a simulation software before ordering the right components such as capacitors, resistors, and even the microcontroller number/type. The best recommendations to complete this project or any future project is to follow three steps. First, select a solution matrix for each important part of the project and see which one will satisfy the project and the sponsor whether from the cost, time, and operation. Second, troubleshooting each component before building it to prevent waste of money, time, and damages. Third, having the project reviewed several times to insure there is no miscalculation. Last but not least, this project objectives were met successfully and our sponsor, who is Terry Walden, received his Window with full satisfaction and happiness since all the requirement were functioned accurately.

APPENDIXES

```
// PIC16F877A Configuration Bit Settings

// 'C' source line config statements

// CONFIG
#pragma config FOSC = HS        // Oscillator Selection bits (HS oscillator)
#pragma config WDTE = OFF       // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF      // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = ON       // Brown-out Reset Enable bit (BOR enabled)
#pragma config LVP = OFF        // Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable
bit (RB3 is digital I/O, HV on MCLR must be used for programming)
#pragma config CPD = OFF        // Data EEPROM Memory Code Protection bit (Data EEPROM code
protection off)
#pragma config WRT = OFF        // Flash Program Memory Write Enable bits (Write protection off; all
program memory may be written to by EECON control)
#pragma config CP = OFF         // Flash Program Memory Code Protection bit (Code protection off)

#define _XTAL_FREQ 20000000 //Specify the XTAL crystal FREQ

Void main ()
{

  TRISB = 0x00; // Set PortB as output to cennect them to Motor
            // RB0 to control the steps.
            // RB1 to control the direction, High moves CW and Low moves CCW
            // RB2 to control the Enable, High to stop the motor, Low will move the motor.
  PORTB = 0x00; // Initialized to be LOW
  TRISC = 0x01; // Set PortC as Input, Portc0 is A channel of re`1mote.
            // Set PortC as Input, Portc1 is B channel of remote.
            // Set PortC as Input, Portc2 is D channel of remote.
            // Set PortC as Input, Portc3 is C channel of remote.
            // Set PortC as Input, Portc4 is to check the sensor when the window is fully opened.
            // Set PortC as Input, Portc5 is to check the sensor when the window is fully Closed.
  PORTC = 0xF0; // set the first four bits of PortC as Low.
            // set the last four bits of PortC as High.

  If (RC0 == 1) // If it receive a function from A channel
  {
    while(RC0 == 1) // While A channel is pressed.
    {
```

```
    if(RC4 == 1)   // check sensor.
    {
       PORTB = 0x01;
       __delay_ms(0.3);
    }
    else
    {
       PORTB = 0x04;
       __delay_ms(1000);
       PORTB = 0x00;
    }
    if(RC4 ==1)
    {
       PORTB = 0x00;
      __delay_ms(0.3);
    }
    else
    {
       PORTB = 0x04;
       __delay_ms(1000);
       PORTB = 0x00;
    }
   }
}

if (RC2 == 1) // if channel B is pressed.
{
   int Count1 = 0;
   while (Count1 <20000)// while loop for fully opening
   {
       if(RC4 ==1&& RC1 ==0 && RC0==0) // check sensor
       {
         PORTB = 0x01;
         __delay_ms(0.3);
       }
       else
       {
          Count1 = 20000;
          PORTB = 0x04;
          __delay_ms(1000);
          PORTB = 0x00;
       }
       if(RC4 ==1 && RC1 ==0 && RC0==0)
       {
```

```
            PORTB = 0x00;
            __delay_ms(0.3);
        }
        else
        {
            Count1 = 20000;
            PORTB = 0x04;
            __delay_ms(1000);
            PORTB = 0x00;
        }
        Count1++;
    }
}

if (RC1== 1) // if channel C is pressed
{
   while(RC1 == 1) // while channel C is pressed.
   {
     if(RC5 == 1)   // check sensor.
     {
        PORTB = 0x03;
     __delay_ms(0.3);
     }
     else
     {
        PORTB = 0x04;
        __delay_ms(1000);
        PORTB = 0x00;
     }
     if(RC5 ==1)
     {
        PORTB = 0x02;
        __delay_ms(0.3);
     }
     else
     {
        PORTB = 0x04;
        __delay_ms(1000);
        PORTB = 0x00;
     }
   }
}

if(RC3 == 1) // if channel D is pressed
```

```
{
   int Count2 =0;
   while(Count2 < 20000) // while loop for fully closing.
   {
    if(RC5 == 1&& RC1 ==0 && RC0==0)  // check sensor.
    {
       PORTB = 0x03;
    __delay_ms(0.3);
    }
    else
    {
       Count2 = 20000;
       PORTB = 0x04;
       __delay_ms(1000);
       PORTB = 0x00;
    }
    if(RC5 ==1 && RC1 ==0 && RC0==0)
    {
       PORTB = 0x02;
       __delay_ms(0.3);
    }
    else
    {
       Count2 = 20000;
       PORTB = 0x04;
       __delay_ms(1000);
       PORTB = 0x00;
    }
    Count2++;
   }
 }

}
```